Assignment 5 64060

Nate Cvelbar

2023-11-30

```
#install.packages("caret")
library(caret)

#install.packages("ISLR") # only install if needed
library(ISLR)

#install.packages("tidyverse") # only install if needed
library(tidyverse)
#install.packages("cluster") # only install if needed
library(cluster)
#install.packages("factoextra") # only install if needed
library(factoextra)
#install.packages("NbClust") # only install if needed
library(NbClust)


#install.packages("stats") # only install if needed
library(stats)

#install.packages("cluster") # only install if needed
library(cluster)

#install.packages("fpc") # only install if needed
library(fpc)
```

```
#Assignment 5
#Nate Cvelbar
#BA-64060

#File taken online from course Assignment 5
#Loading the dataset
cereals=read.csv('C:/Users/Owner/Documents/Cereals.csv')
```

```
#Remove missing values
cereals <- na.omit(cereals)
head(cereals)
```

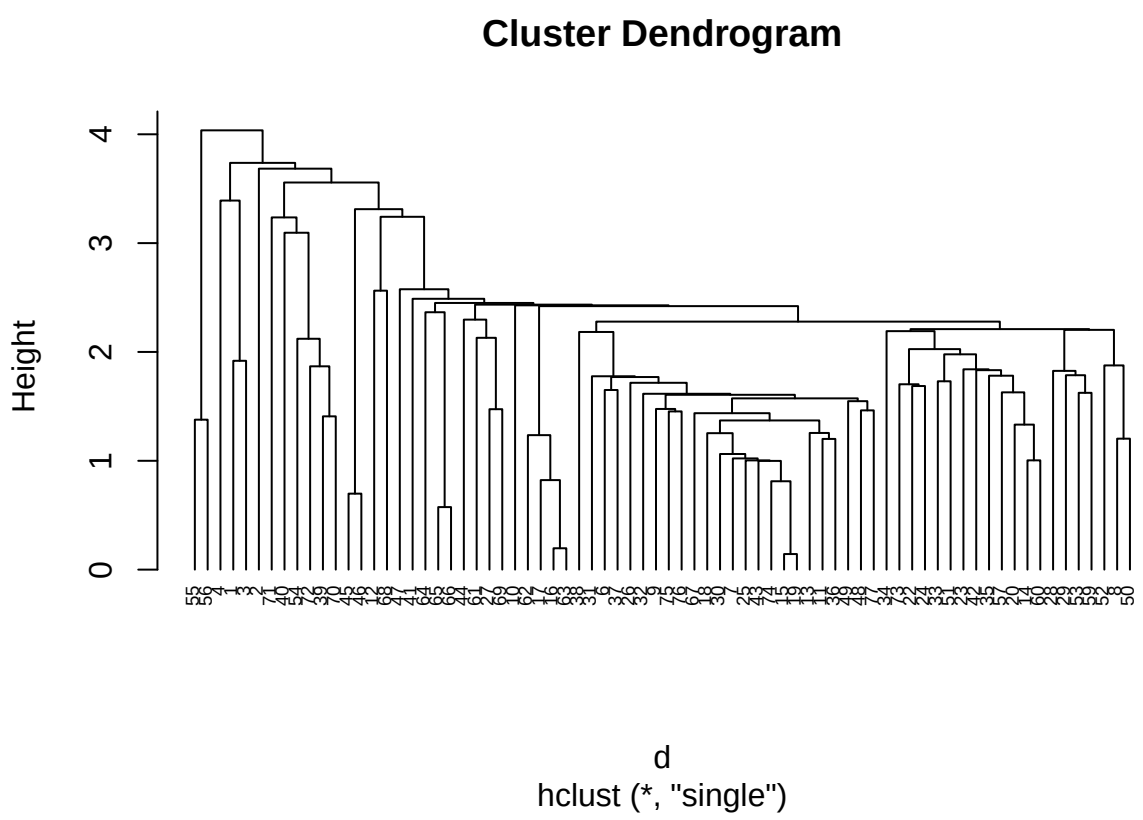
```
##                        name mfr type calories protein fat sodium fiber carbo
## 1                 100%_Bran   N    C       70       4   1    130  10.0   5.0
## 2         100%_Natural_Bran   Q    C      120       3   5     15   2.0   8.0
## 3                  All-Bran   K    C       70       4   1    260   9.0   7.0
```

```
## 4 All-Bran_with_Extra_Fiber   K    C       50       4   0    140  14.0   8.0
## 6   Apple_Cinnamon_Cheerios   G    C      110       2   2    180   1.5  10.5
## 7                Apple_Jacks   K    C      110       2   0    125   1.0  11.0
##   sugars potass vitamins shelf weight cups   rating
## 1      6    280       25     3      1 0.33 68.40297
## 2      8    135        0     3      1 1.00 33.98368
## 3      5    320       25     3      1 0.33 59.42551
## 4      0    330       25     3      1 0.50 93.70491
## 6     10     70       25     1      1 0.75 29.50954
## 7     14     30       25     2      1 1.00 33.17409
```

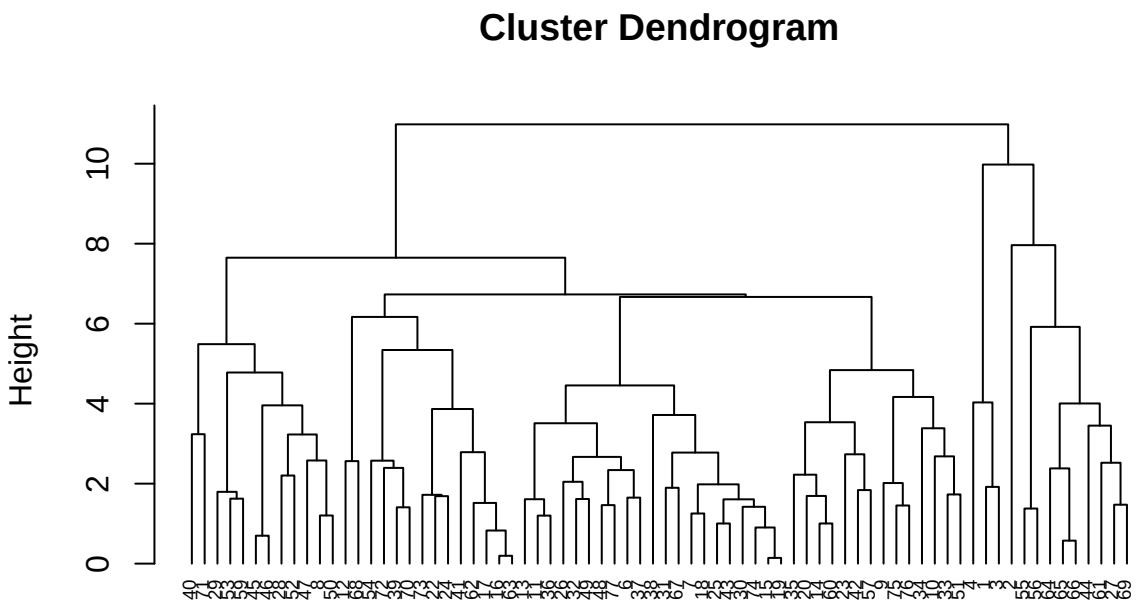
```
#Remove the name, mfr, and type columns for now since they are alphabetic. Can reference alphaC later.
alphaC = cereals[c(1:3)]
cereals <- cereals[-c(1:3)]

#Scale the data
cereals <- scale(cereals)
head(cereals)
```

```
##     calories    protein        fat     sodium       fiber      carbo     sugars
## 1 -1.8659155  1.3817478  0.0000000 -0.3910227  3.22866747 -2.5001396 -0.2542051
## 2  0.6537514  0.4522084  3.9728810 -1.7804186 -0.07249167 -1.7292632  0.2046041
## 3 -1.8659155  1.3817478  0.0000000  1.1795987  2.81602258 -1.9862220 -0.4836096
## 4 -2.8737823  1.3817478 -0.9932203 -0.2702057  4.87924705 -1.7292632 -1.6306324
## 6  0.1498180 -0.4773310  0.9932203  0.2130625 -0.27881412 -1.0868662  0.6634132
## 7  0.1498180 -0.4773310 -0.9932203 -0.4514312 -0.48513656 -0.9583868  1.5810314
##       potass   vitamins      shelf     weight       cups     rating
## 1  2.5605229 -0.1818422  0.9419715 -0.2008324 -2.0856582  1.8549038
## 2  0.5147738 -1.3032024  0.9419715 -0.2008324  0.7567534 -0.5977113
## 3  3.1248675 -0.1818422  0.9419715 -0.2008324 -2.0856582  1.2151965
## 4  3.2659536 -0.1818422  0.9419715 -0.2008324 -1.3644493  3.6578436
## 6 -0.4022862 -0.1818422 -1.4616799 -0.2008324 -0.3038480 -0.9165248
## 7 -0.9666308 -0.1818422 -0.2598542 -0.2008324  0.7567534 -0.6553998
```

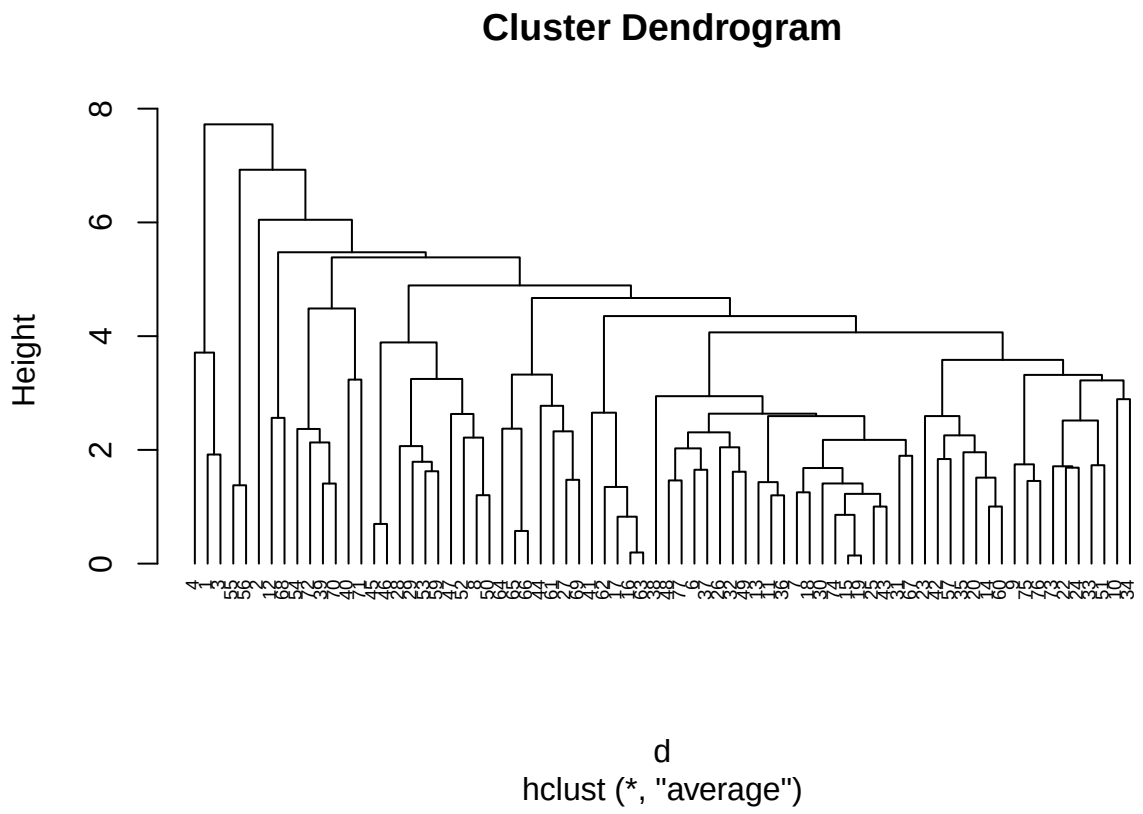
```
#cereals <-cbind(alphaC,cereals)
#head(cereals)
#Apply Euclidean distance to the normalized measurements
d <- dist(cereals, method="euclidean")
#Generate different clusters from the 4 specified methods
hc1 <- hclust(d, method = "single" )
hc2 <- hclust(d, method = "complete" )
hc3 <- hclust(d, method = "average" )
hc4 <- hclust(d, method = "ward" )

plot(hc1, cex = 0.6, hang = -1)
```



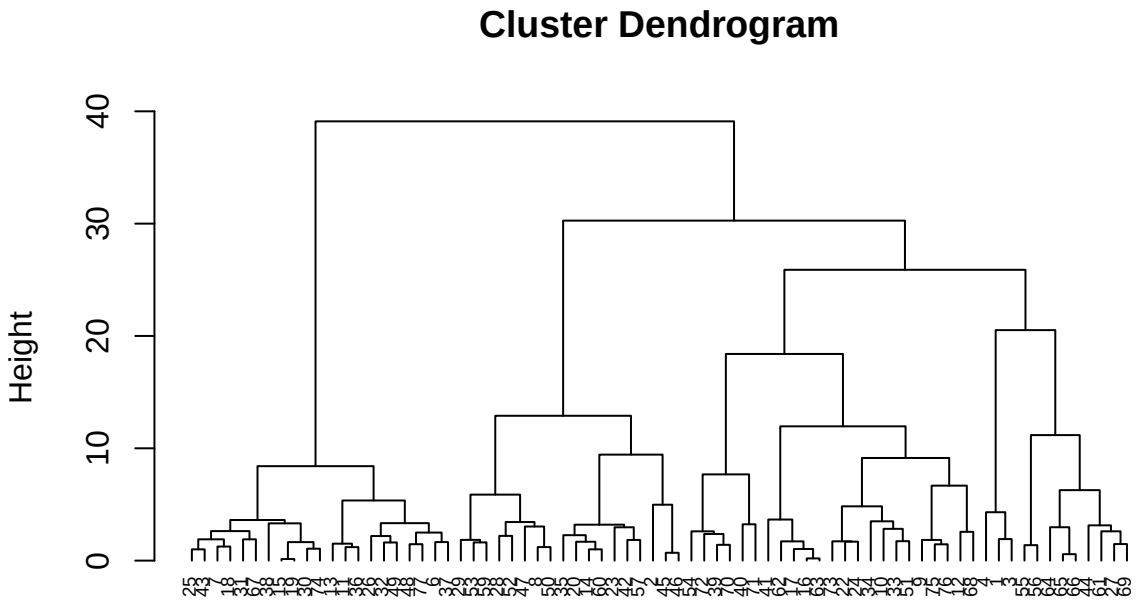

```
plot(hc2, cex = 0.6, hang = -1)
```




```
plot(hc3, cex = 0.6, hang = -1)
```




```
plot(hc4, cex = 0.6, hang = -1)
```




```
#Based on these dendograms, I found cluster numbers of 10, 3, 3, and 4.
#Based on these results, my initial guess for the number of clusters is 4.
#I will now use the Agnes function to look further into this.
```

```
#Agnes function
hc_single <- agnes(cereals, method = "single")
hc_complete <- agnes(cereals, method = "complete")
hc_average <- agnes(cereals, method = "average")
hc_ward <- agnes(cereals, method = "ward")

#Comparing agglomerative coefficients
print(hc_single$ac)
```

```
## [1] 0.6067859
```

```
print(hc_complete$ac)
```

```
## [1] 0.8353712
```

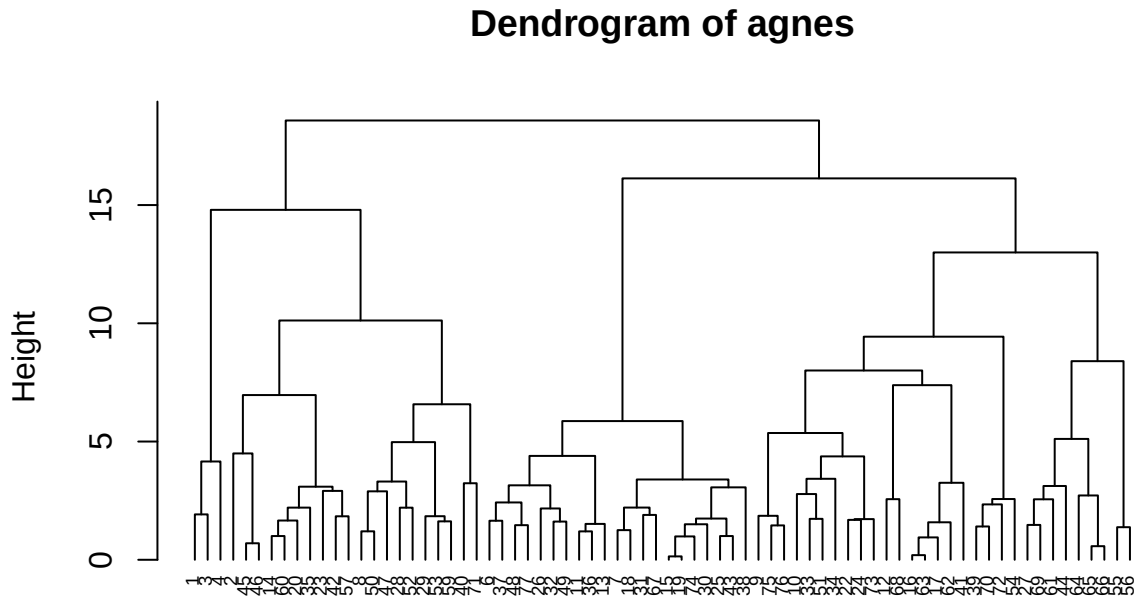
```
print(hc_average$ac)
```

```
## [1] 0.7766075
```

```r
print(hc_ward$ac)
```

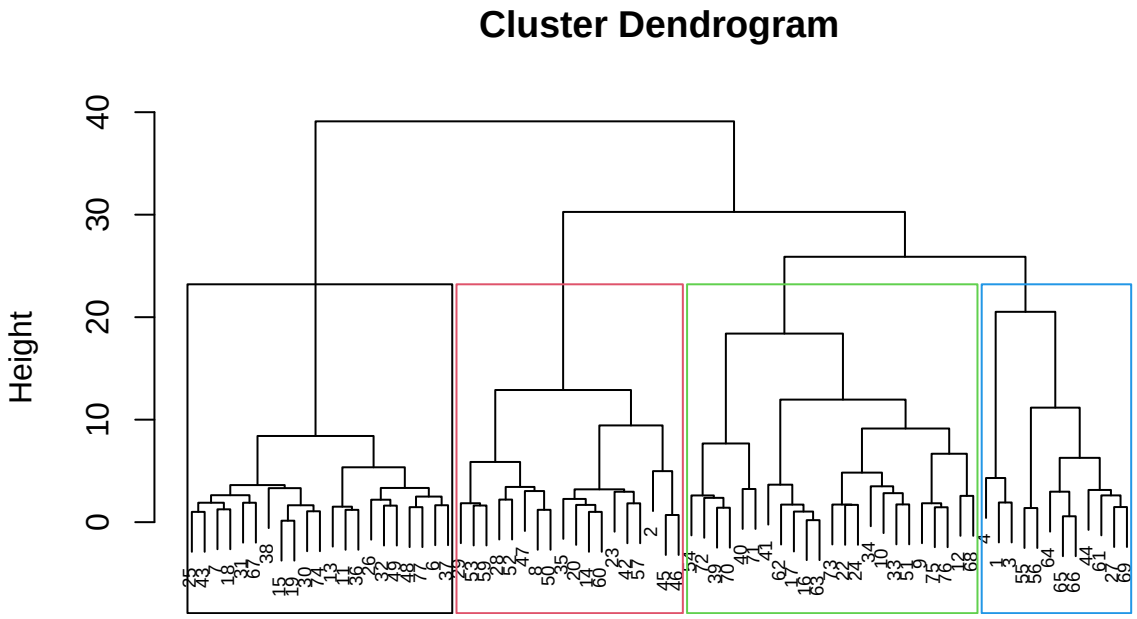
```
## [1] 0.9046042
```

```r
#Based on Agnes, the best method is the ward method, since its ac is highest
pltree(hc_ward, cex = 0.6, hang = -1, main = "Dendrogram of agnes")
```




```r
#Based on all the data collected, and the various dendograms and ac values, I will stick with my initia
```

```r
#Divide the data into the clusters
hc_ward1 <- hclust(d,method = "ward")
# plot dendrogram to show clusters
plot(hc_ward1, cex = 0.6)
rect.hclust(hc_ward1, k = 4, border = 1:4)
```




```
#Split into clusters
clusters = cutree((hc4), k=4)
clust.centroid = function(i, dat, clusters) {
    ind = (clusters == i)
    colMeans(dat[ind,])
}

sapply(unique(clusters), clust.centroid, cereals, clusters)
```

```
##                [,1]       [,2]       [,3]         [,4]
## calories -1.48796546  0.7657366  0.1978117 -0.003552997
## protein   0.29728513  0.6071316 -0.9199689  0.209719819
## fat      -0.74491520  1.0483992  0.0000000 -0.431834896
## sodium   -1.41293342 -0.1225404  0.1210114  0.722595168
## fiber     0.89034641  0.3860027 -0.6619844 -0.162197082
## carbo    -0.40164279 -0.3017144 -0.5423583  0.940873779
## sugars   -1.01888691  0.4594980  0.9583619 -0.703040058
## potass    0.66173851  0.7303221 -0.7415648 -0.239730396
## vitamins -0.64907563 -0.2441400 -0.1818422  0.695744074
## shelf     0.04060222  0.8084353 -0.6604628 -0.050841040
## weight   -0.83636138  0.7587982 -0.2008324  0.025889154
## cups     -0.37455471 -0.6338128  0.2779676  0.437650729
## rating    1.66407332 -0.2475159 -0.9636465  0.205347137
```

```
#We know that cluster 3 contains approximately 30% of data
#Based on the information found here, we can conclude that the clusters are indeed stable, since the pa
```

```
cereals1=read.csv('C:/Users/Owner/Documents/Cereals.csv')
cereals1 <- na.omit(cereals1)
#Elementary school questions
#If we manually split out the clusters to show the proper information for each, they look as follows:
#Manually split
cluster1=cereals1[-c(58,21,29,53,59,28,52,47,8,50,35,20,14,60,23,42,57,2,45,46,54,72,39,70,40,71,41,62,
cluster2=cereals1[-c(21,58,5,25,43,7,18,31,67,38,15,19,30,74,13,11,36,26,32,49,48,77,6,37,54,72,39,70,4
cluster3=cereals1[-c(5,25,58,43,7,18,31,67,38,15,19,30,74,13,11,36,26,32,49,48,77,6,37,29,53,59,28,52,4
cluster4=cereals1[-c(21,5,25,43,7,18,31,67,38,15,19,30,74,13,11,36,26,32,49,48,77,6,37,29,53,59,28,52,4

#To find a suitable solution for the school, we must have a set with at least 5 cereals (each for a dif
#This means having cereal low in sugar and carbs, but high in vitamins, fiber, and protein.
#We could perform another analysis with greater emphasis on these values, but the current clusters alre
#It is not necessary to de-scale the data first, but it does make things easier to read
#As for normalization, I think it is more intuitive to read the raw data here, though of course the clu
#Based on the specs I layed out, cluster 4 here is a great choice for the school. It checks all the box
```